# DØ Data Reprocessing at GridKa

- Procedures
- Book-keeping
- Statistics
- Problems
- Suggestions for improvements

## Procedures I

- Import of DST data from FNAL in advance
  - needed for stability and efficiency
    - to avoid dying jobs due to delivery problems/timeouts
    - not wasting CPU resources during wait time
  - semi-automatic, generic shell scripts using only list of datasets as input
- Job Submission:
  - semi-automatic, generic shell scripts using only list of datasets as input
    - automatic creation of macro etc.
  - file-input handled by SAM
  - I/O with central GFPS file server (parallel system, IBM)

## Procedures II

Book-keeping

- simple-minded (shell-scripts, no DB), partly using SAM
- semi-automatic: generic shell scripts using only list of submitted datasets as input
  - incorporates SAM commands
  - checks:
    - existence of all necessary files (e.g. tagfile)
    - absence of error files
    - ’closed’ status for events.read/write file
  - automatic creation SAM datasets for failed jobs
    - comparing dataset filelist (sam translate constraints...) with job output files
    - based on own book-keeping, not relying on processing status in SAM
    - creates list of datasets for failed jobs, which is passed to job submission

## Procedures III

- – several iterations until failures disappear or manual check in case of remaining problematic files
    - ∗ most failures were due to non-existence of file locations in SAM suggestion: include ’availability_status available’ in dataset definition
  - have not checked for file corruption
    - ∗ 6 corrupted TMBs were only detected at FNAL, which were already corrupted locally (mostly due to NFS errors)

- TMB import to FNAL, local storing of DST output (Daniel W.)
  - after subsets of data have been declared as ’completely processed’
  - incorporates own book-keeping

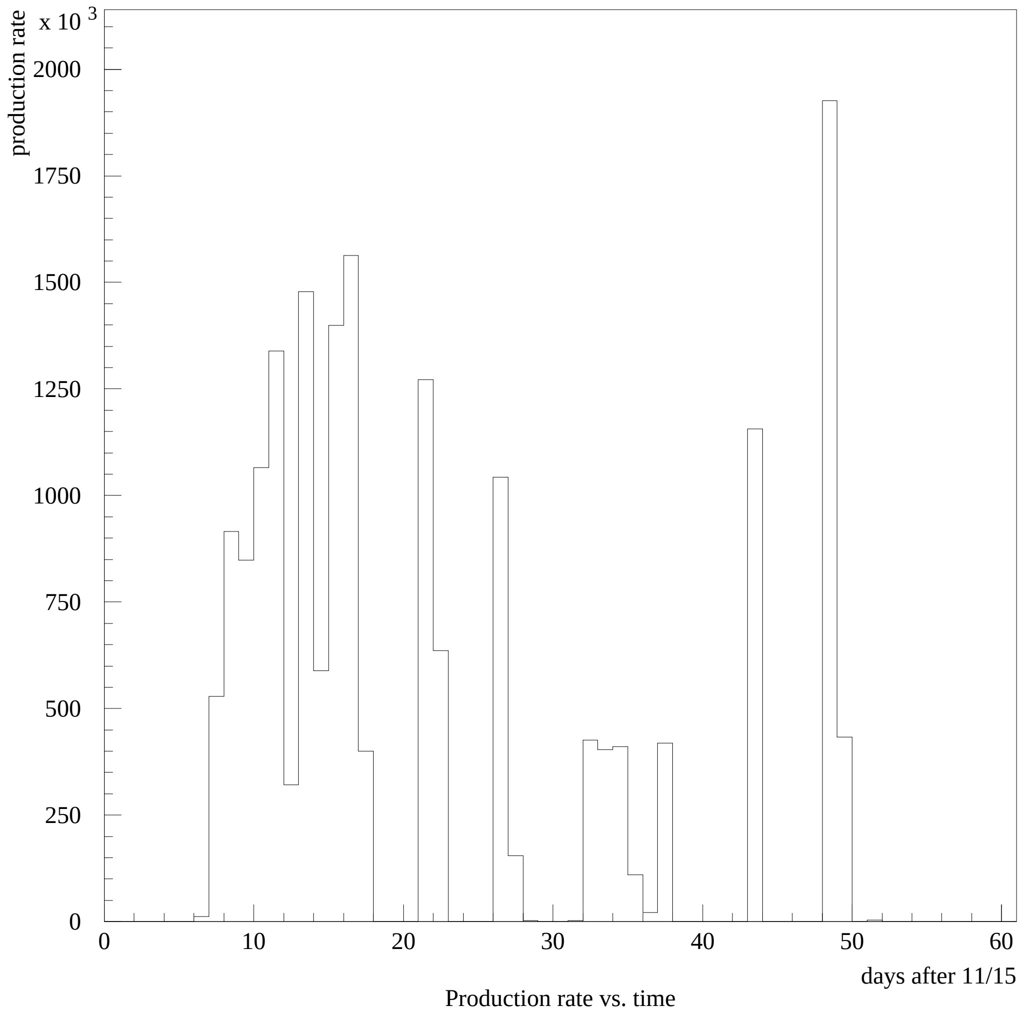
## Procedures IV: Criticism

- only partly automated, expert level scripts
- do not have a local DB for sub submission
- high rate of job failures (see below) and subsequent diagnostics made it impossible to completely automate job submission/book-keeping
  - But: only site with diagnosis for every single failed job (see below)
- manpower intensive due to high rate of failures/problems

## Reprocessed Sample

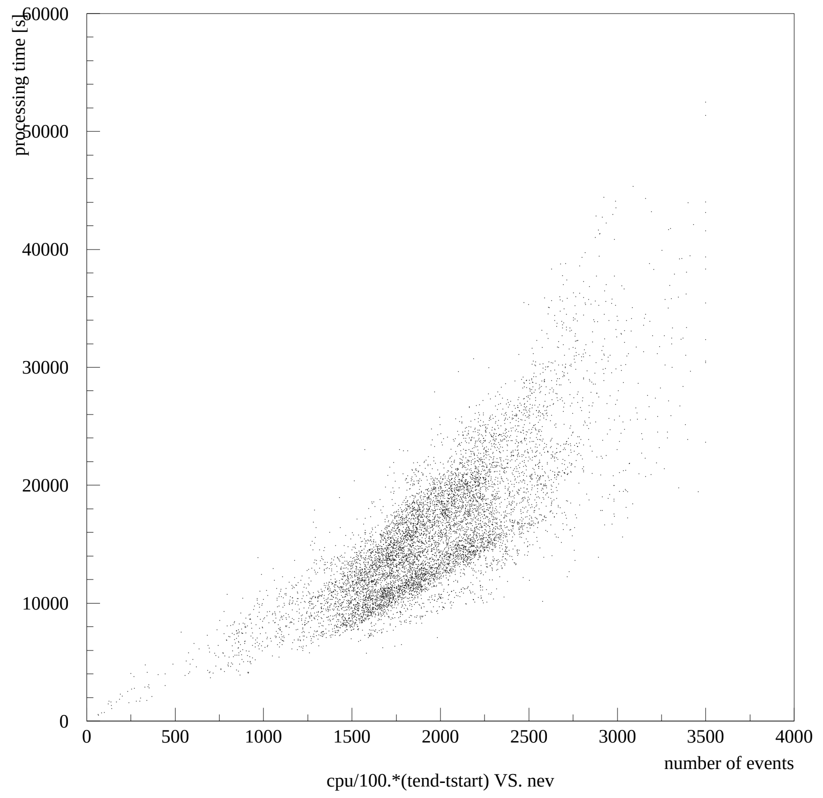
- 10600 of 10662 assigned files processed successfully (failure rate: 0.6%)
  - containing ∼21M events in total
  - 62 failed jobs (files)
    - ∗ 50 files with status non-available (no location declared in SAM)
    - ∗ 9 files on bad tape (status not allowed)
    - ∗ 1 genuine reco crash (evpack checksum test)
    - ∗ 2 files with non-reconstructable event
- data import rate: 2 MB/s effective
- CPUs available: 50-300 CPUs
  - average: 150 in absence of technical problems
  - 1 CPU at GridKa corresponds on average to 2.3 GHz

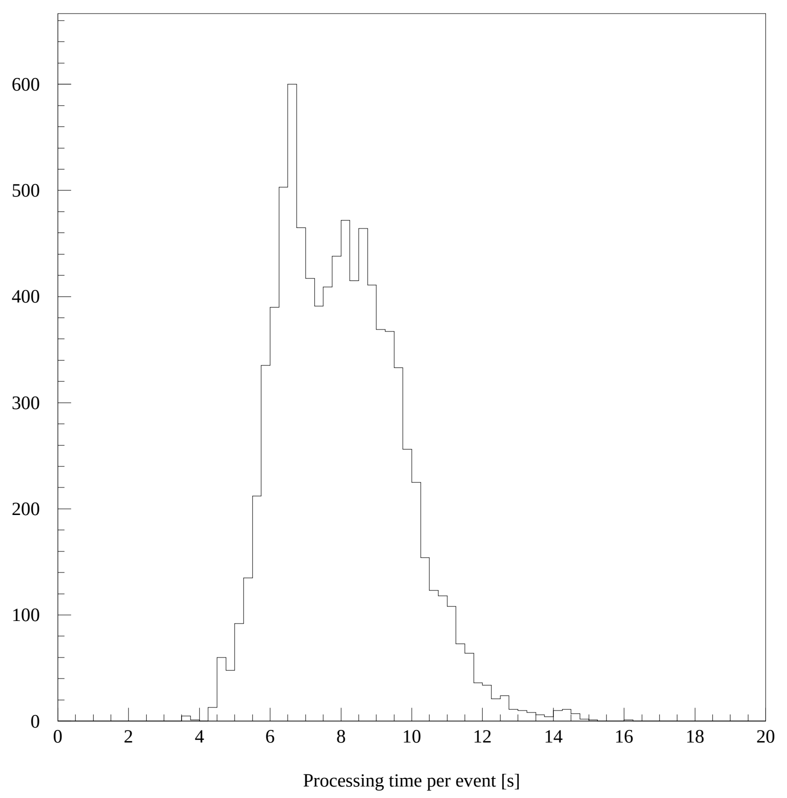
## Statistics I: Production Rate



- $>$ 1M events per day if stable conditions and resources allow

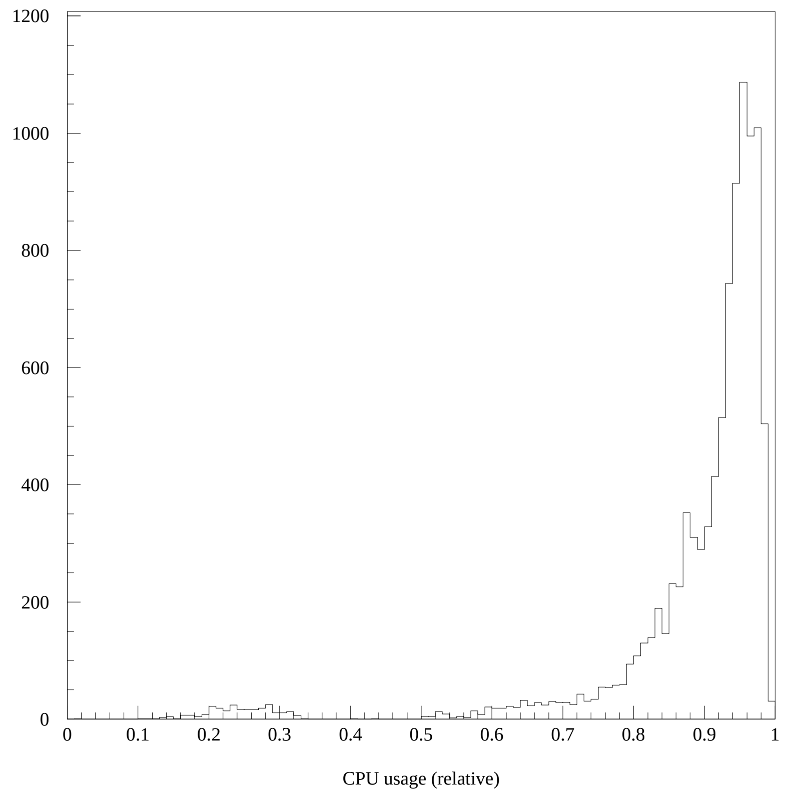
## Statistics II: Processing Time

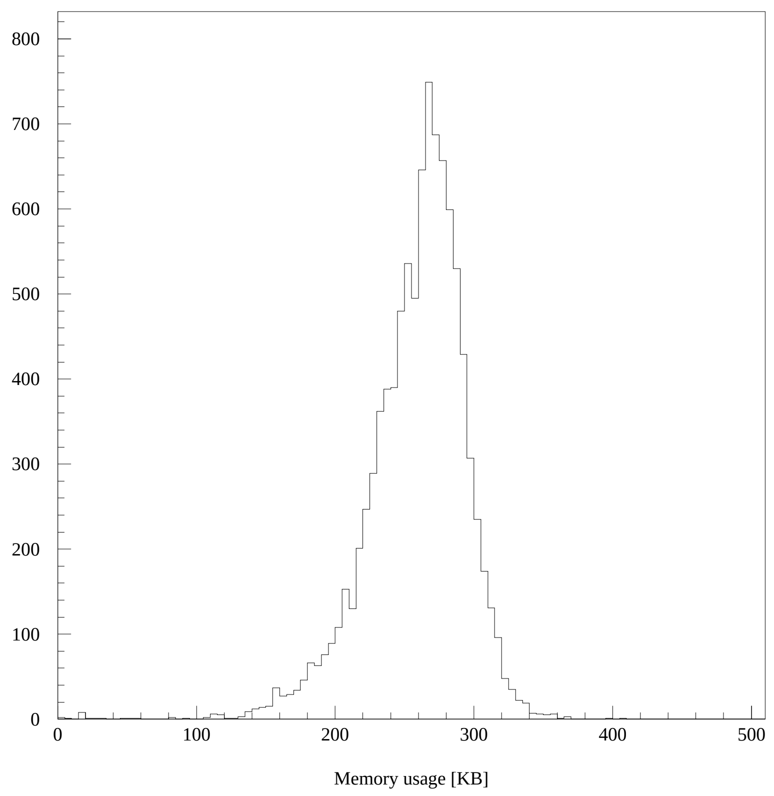




- Positive curvature: real or artefact of correlation between number of events is run and luminosity?
- $\sim$8 s processing time per event on 2.3 GHz (average) node

## Statistics III: CPU and Memory Consumption





- GFPS filesystem can feed $>$300 running parallel jobs without I/O limitations.
- No tails to large memory consumption seen as in previous reco releases

## Problems I: Global Ones

- data import
  - bottleneck enstore: situation improved significantly with rp-router
  - delays in run assignment (lost several days with idling farm)
- SAM
  - local and remote SAM uptime
    - several hundreds jobs lost due to SAM downtime
  - data import down for several days
  - possibility to prestage file to local disk of worker node (non-SAM) missing

## Problems II: Local Ones

- Local problems at GridKa were far more frequent and serious:
  - 2050/12650 failed jobs (16%)
    - makes an completely automated job submission close to impossible
  - NFS problems
  - PBS errors (improved after switch to PBSpro)
  - Problems with GFPS file server
    - hardware and software problems, inode limits
    - Note: During stable copnditions GFPS showed very good performace, could feed $>$300 jobs with input data, without significant delay

## Improvements Needed for Next Round

- 1st: stability/reliability of GridKa farm!
- SAM stability
  - prefer to keep file handling within SAM
- suggest central collection of submission/utility scripts (need to be made user friendly)
  - DB option (cf. Lyon) of advantage, but not realized at most farms
- check and merging at processing sites (need to develop software for that)
  - Note: Even for MC, merging and SAM declaring/storing can seldom be done automatically, due to failed/crashed jobs.
- Local data base proxy for reprocessing from RAW
  - deployed and extensively tested at GridKa
  - requirement for farms on local networks
  - performance: DB access/wait time reduced by factor $\sim$ 15
  - stability: CORBA communication failures possible for remote access, results in job crashes.